

```

#!/bin/sh
#=====
# Xen vlan bridge start/stop script.
# Xend calls a network script when it starts.
# The script name to use is defined in /etc/xen/xend-config.sxp
# in the network-script field.
#
# This script creates a bridge (default vlandev${vlan}), creates a device
# (default eth0.${vlan}), and adds it to the bridge. This scrip assumes
# the Dom0 does not have an active interface on the selected vlan; if
# it does the network-bridge script should be used instead.
#
# To use this script, vconfig must be installed.
#
# Usage:
#
# network-bridge-vlan (start|stop|status) {VAR=VAL}*
#
# Vars:
#
# vlan          The vlan to bridge (default 2)
# bridge        The bridge to use (default vlandev${vlan}).
# netdev        The interface to add to the bridge (default eth0}).
#
# Internal Vars:
# vlandev="${netdev}.${vlan}"
#
# start:
# Creates the bridge
# Adds vlandev to netdev
# Enslaves vlandev to bridge
#
# stop:
# Removes vlandev from the bridge
# Removes vlandev from netdev
# Deletes bridge
#
# status:
# Print vlan, bridge
#
#=====

dir=$(dirname "$0")
. "$dir/xen-script-common.sh"

findCommand "$@"
evalVariables "$@"

vlan=${vlan:-2}
bridge=${bridge:-vlandev${vlan}}
netdev=${netdev:-eth0}

vlandev="${netdev}.${vlan}"

##
# link_exists interface
#
# Returns 0 if the interface named exists (whether up or down), 1 otherwise.
#
link_exists()
{
    if ip link show "$1" >/dev/null 2>/dev/null
    then

```

```

        return 0
    else
        return 1
    fi
}

# Usage: create_bridge bridge
create_bridge () {
    local bridge=$1

    # Don't create the bridge if it already exists.
    if ! brctl show | grep -q ${bridge} ; then
        brctl addbr ${bridge}
        brctl stp ${bridge} off
        brctl setfd ${bridge} 0
    fi
    ip link set ${bridge} up
}

# Usage: add_to_bridge bridge dev
add_to_bridge () {
    local bridge=$1
    local dev=$2
    # Don't add $dev to $bridge if it's already on a bridge.
    if ! brctl show | grep -q ${dev} ; then
        brctl addif ${bridge} ${dev}
    fi
}

# Usage: show_status vlandev bridge
# Print vlan and bridge
show_status () {
    local vlandev=$1
    local bridge=$2

    echo '=====
cat /proc/net/vlan/${vlandev}
echo ' '
brctl show ${bridge}
echo '=====
}

op_start () {
    if [ "${bridge}" = "null" ] ; then
        return
    fi

    if ! link_exists "$netdev"; then
        return
    fi

    if link_exists "$vlandev"; then
        # The device is already up.
        return
    fi

    create_bridge ${bridge}

    ip link set ${netdev} up

    vconfig set_name_type DEV_PLUS_VID_NO_PAD
    vconfig add ${netdev} ${vlan}
    ip link set ${vlandev} address fe:ff:ff:ff:ff:ff

```

```

    ip link set ${vlandev} up
    ip link set ${bridge} up

    add_to_bridge2 ${bridge} ${vlandev}
}

op_stop () {
    if [ "${bridge}" = "null" ]; then
        return
    fi
    if ! link_exists "$bridge"; then
        return
    fi

    if link_exists "$vlandev"; then
        ip link set ${vlandev} down

        brctl delif ${bridge} ${vlandev}
        ip link set ${bridge} down

        vconfig rem ${vlandev}
    fi
    brctl delbr ${bridge}
}

# adds $dev to $bridge but waits for $dev to be in running state first
add_to_bridge2() {
    local bridge=$1
    local dev=$2
    local maxtries=10

    echo -n "Waiting for ${dev} to negotiate link."
    for i in `seq ${maxtries}` ; do
        if ifconfig ${dev} | grep -q RUNNING ; then
            break
        else
            echo -n '.'
            sleep 1
        fi
    done

    if [ ${i} -eq ${maxtries} ] ; then echo '(link isnt in running state)' ; fi

    add_to_bridge ${bridge} ${dev}
}

case "$command" in
    start)
        op_start
        ;;

    stop)
        op_stop
        ;;

    status)
        show_status ${vlandev} ${bridge}
        ;;

    *)
        echo "Unknown command: $command" >&2
        echo 'Valid commands are: start, stop, status' >&2
        exit 1
esac

```